

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
Факультет кібербезпеки, комп'ютерної та програмної інженерії
Кафедра комп'ютерних інформаційних технологій

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач випускової кафедри
_____ А.С. Савченко
« ____ » _____ 20__ р

ДИПЛОМНИЙ ПРОЕКТ

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ «БАКАЛАВР»

Тема: «Online ресторан на основі WEB - технології»

Виконавець: студентка УС-412 Сохань Юлія Сергіївна
(студент, група, прізвище, ім'я, по батькові)

Керівник: к. т. н., доцент Климова Асія Сабирівна
(науковий ступень, вчене звання, прізвище, ім'я, по батькові)

Нормоконтролер: ст. викл. Шевченко О.П.
(П.І.Б.) (підпис)

КИЇВ 2021

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп'ютерної та програмної інженерії

Кафедра комп'ютерних інформаційних технологій

Освітній ступінь: Бакалавр

Галузь знань, спеціальність, спеціалізація: 12 “Інформаційні технології”,
122 “Комп'ютерні науки”, “Інформаційні управляючі системи та технології”.

ЗАТВЕРДЖУЮ

Завідувач кафедри

А.С. Савченко

“ ____ ” ____ 2021 р.

ЗАВДАННЯ

на виконання дипломного проекту студента

Сохань Юлія Сергіївна

(прізвище, ім'я, по батькові)

1. Тема проекту: «Online ресторан на основі WEB - технології» затверджена наказом ректора № 636/ст. від 22.04.2021р.
2. Термін виконання роботи: з 10.05.2021 по 20.06.2021р.
3. Вихідні дані до роботи: розробка веб-сайту на основі мови програмування JavaScript.
4. Зміст пояснювальної записки (перелік питань, що підлягають розробці): вступ, теоретична частина та постановка задачі, дослідження технологій та засобів, розробка програмного продукту, висновки.
5. Перелік обов'язкового графічного матеріалу: загальний перелік архітектури серверної частини у середовищі розробки. Перелік етапів розробки програмного продукту.

КАЛЕНДАРНИЙ ПЛАН

	Етапи виконання дипломної роботи	Термін виконання етапів	Примітка
1	Аналіз літератури та джерел за темою дипломного проекту.	11.05.2021р. – 12.05.2021р.	
2	Розробка та затвердження плану дипломного проекту.	13.05.2021р.	
3	Проведення консультації з науковим керівником щодо створення першого розділ.	14.05.2021р.	
4	Аналітичний огляд і постановка задачі.	15.05.2021р. – 18.05.2021р.	
5	Аналіз теорії та засобів розробки.	19.05.2021р. – 22.05.2021р.	
6	Створення серверної частини.	23.06.2021р. – 27.05.2021р.	
6	Розробка Online Ресторану на основі WEB – технології.	28.05.2021р. – 04.06.2021р.	
7	Висновки та оформлення пояснювальної записки дипломного проекту.	05.06.2021р. – 08.06.2021р.	
8	Підписання необхідних документів у встановленому порядку.	09.06.2021р. – 10.06.2021р.	
9	Підготовка до захисту та попередній захист дипломного проекту на випусковій кафедрі дипломного проекту	11.06.2021р. – 12.06.2021р.	

Студент

(*Сохань Ю.С.*)

Керівник дипломної роботи

(*Климова А.С.*)

РЕФЕРАТ

Пояснювальна записка до дипломного проекту «Online ресторан на основі WEB – технології» містить: 77 сторінок, 17 рисунків, 8 літературних джерел.

Об'єкт дослідження: WEB - технології.

Предмет дослідження: Online ресторан на основі WEB - технології.

Мета роботи: створення веб-сайту ресторана, який надає можливість користувачам переглядати інформацію ресторана, меню, ціни та бронювати стіл у зручний для них час.

Методи дослідження, технічні та програмні засоби: засоби реалізації, порівняльний аналіз, обробка літературних джерел.

Отримані результати та їх новизна: було розроблено веб-сайт ресторану, який надає можливість, переглядати інформацію про ресторан, його контактні данні, місце знаходження, години роботи, опис самого ресторану, його меню, ціни на страви, бронювати стіл, переглядати галерею, читати відгуки та підписуватись, щоб отримувати ексклюзивні рецепти від шеф-кухаря. Проаналізовано існуючі веб-сати, визначено їх переваги та недоліки. Проаналізовано вимоги до нового веб-сайту ресторана. З використанням бібліотеки React.js та мови програмування JavaScript реалізовано графічний веб-сайт ресторану та створено Node.js сервер, для взаємодії з клієнтами.

ONLINE РЕСТОРАН, ВЕБ-САЙТ, МЕНЮ, ІНФОРМАЦІЯ ПРО РЕСТОРАН,
БРОНЮВАННЯ, СЛОВНИК

.

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ ТА ТЕОРЕТИЧНА ЧАСТИНА.....	9
1.1. Основні питання та визначення теорії надійності	9
1.2. Відмова та її види	10
1.3. Види показників надійності	11
1.4. Класифікація Веб-сайтів.....	14
1.6. HTML – мова розмітки гіпертексту	18
1.7. CSS – каскадні таблиці стилей.....	20
1.8. Bootstrap – інструмент з шаблонами CSS та HTML	20
1.10. Постановка задачі.....	22
Висновки до розділу 1	23
РОЗДІЛ 2. ЗАСОБИ РЕАЛІЗАЦІЇ ТА РОЗРОБКИ.....	24
2.1. Пакетний менеджер npm	24
2.2. Мова програмування JavaScript.....	25
2.2.1. JavaScript на стороні клієнта	26
2.2.2. Переваги JavaScript	26
2.2.3. Обмеження JavaScript	27
2.3. Бібліотека реалізації веб-інтерфейсу React.js.....	28
2.4. Віртуальний DOM	29
2.5. Single Page Application	30
2.5.1. Основні елементи SPA.....	31
2.5.2. Переваги і недоліки SPA	31
2.5.3. Коли використовувати SPA.....	33
2.6. Створення веб-сайту за допомогою React.js	33
2.7. Створення серверної частини з Express.js	34
2.8. Середовище розробки Visual Studio Code	36
2.8.1. Особливості мови в VS Code.	37
Висновки до розділу 2	38

РОЗДІЛ 3. ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ	39
3.1. Загальний опис роботи веб-сайту.....	46
Висновки до розділу 3	47
СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ.....	49
ДОДАТКИ.....	50
Додаток А	50

ВСТУП

В останні роки комп'ютери та програмне забезпечення широко використовуються у всіх сферах життя. Насправді, запорукою засвоєння великої кількості знань та навичок є вміння користуватися комп'ютером.

Сьогодні неможливо уявити людську діяльність без використання комп'ютерів, а також використання Інтернету, який надає користувачам всілякі інформаційні та комунікаційні послуги. В Інтернеті можна використовувати такі послуги:

- перегляд вмісту мультимедійних сторінок;
- надсилання та отримання електронних листів;
- спілкування в переговорних кімнатах;
- покупки в магазинах;
- брати участь у телеконференціях, форумах тощо.

Розроблюваний сайт повинен являти собою автоматизований сайт для полегшення отримання користувачами актуальної інформації про ресторан і його діяльності. Він повинен являти собою відкриту, розширену масштабовану і модифіковану систему.

Користувачів інтернету з кожним днем стає все більше. А це означає, що поширення інформації в мережі відбувається дуже швидко. Тепер з легкістю можна замовити їжу онлайн або забронювати столик в ресторані використовуючи мережу інтернет. Тому необхідно розміщувати інформацію не тільки в друкованих джерелах, але і в інтернеті. Web - ресурс необхідний ресторану для надання користувачам мережі і потенційним клієнтам ресторану постійно мінливій інформації. Будь-який відвідувач повинен мати можливість оцінити рівень сервісу в даному закладі не тільки шляхом особистої присутності, але і за допомогою різних доступних йому ресурсів, в тому числі інтернету. Сайт як візитну картку ресторану дозволяє клієнту дізнатися по карті міста, де розміщується заклад, як можна зручно

дістатися, юридична адреса закладу, контактні телефони, а також години роботи і особливі пропозиції, які з'являються в той чи інший період.

РОЗДІЛ 1

ПОСТАНОВКА ЗАДАЧІ ТА ТЕОРЕТИЧНА ЧАСТИНА

1.1. Основні питання та визначення теорії надійності

Надійність та діагностика - одна з головних інженерних проблем, яка до сьогодні не вирішена. Надійність пов'язана із надмірністю, тому певні запаси використовуються для розрахунку проблем інженерної надійності. Причини, пов'язані з проблемою надійності:

- різке збільшення складності сучасної технології, яка включає 10 і 100 мільйонів елементів;
- інтенсивність режимів роботи систем або їх компонентів;
- складність умов експлуатації технічних засобів (низькі або високі температури, висока вологість вібрації, прискорення, випромінювання) автоматичного обладнання використовуються при зміні температур (від -70 до +60 ° С, при відносній вологості 98-100%, за наявності високої сонячної та космічної радіації);
- вимоги до якості технічних засобів (висока точність, ефективність, швидкість);
- підвищення відповідальності за формування технічних засобів (дуже висока технічна та економічна вартість відмов);
- людський фактор.

Кафедра КІТ (47)				НАУ 21 42 93 000 ПЗ			
Виконав	Сохань Ю.С.			ПОСТАНОВКА ЗАДАЧІ ТА ТЕОРЕТИЧНА ЧАСТИНА	Літера	Аркуш	Аркушів
Керівник	Климова А.С.					8	14
Консульт.					412 122		
Н-котрол.	Шевченко О.П.						
Зав. каф.	Савченко А.С.						

Надійність з її методологією, літературою та науковою школою стала окремою галуззю у 49-50-х роках. Саме в цей час спостерігалася тенденція до вивчення несправностей, що виникають в обладнанні, та факторів, що впливають на надійність обладнання.

У пристроях 40-45% поломок виникають через конструктивні помилки, 20% - несправності через виробничі помилки, 30% - помилки в роботі обслуговуючого персоналу, 5-7% - несправності через деградацію матеріалів під час експлуатації та зберігання пристрої та елементи (старіння елементів, відмови у часі) Проблема підвищення надійності є складною. Надійність не виникає спонтанно, вона завжди є результатом упущень, недисциплінованості, які допускаються при проектуванні, виготовленні та експлуатації виробу, недбалості. При зберіганні продуктів слід враховувати надійність. Надійність продукту в більшості випадків згадується як якість. Якість продукції - це сукупність властивостей, що визначають ступінь її придатності для використання за призначенням. Кожен виріб має свої властивості, особливі показники якості, які проявляються в процесі його застосування (вироби - свіжість, взуття - міцність, комфортність прокладки, відповідність моді, машини - продуктивність, точність, швидкість, швидка та легка взаємозамінність). Більшість властивостей, що характеризують якість, не пов'язані між собою.

1.2. Відмова та її види

Помилка - це подія, внаслідок якої об'єкт втрачає здатність виконувати необхідні функції. Несправність і несправність різні: несправність - це умова і причина несправності, а несправність - це подія.

Несправність ресурсу - це несправність, яка призводить до досягнення об'єктом граничного стану.

Несправність - це найбільш виправна (разова) несправність, яку усуває оператор незначними втручаннями. Повторна відмова - це найбільш виправна помилка того самого характеру, яка повторюється неодноразово.

Деградована несправність - це несправність, спричинена процесами деградації на об'єкті, з дотриманням усіх встановлених правил, стандартів виробництва та експлуатації.

Раптовий збій - збій, який неможливо визначити (передбачити) дослідженням, технічним оглядом.

Поступовий збій - збій, спричинений поступовими змінами одного або декількох параметрів об'єкта, його можна передбачити дослідженням або технічним оглядом, а іноді можна запобігти заходами технічного обслуговування.

Слід розрізняти несправності:

- обліковані та невраховані;
- залежні та незалежні.

Враховуються збої - зазвичай вводяться при розрахунку величини коефіцієнта відмов. Виключені збої - вилучені з розрахунку значення частоти відмов.

Залежна відмова - це відмова об'єкта, спричинена прямо чи опосередковано відмовою або несправністю іншого об'єкта.

Незалежна відмова - це відмова, не спричинена прямо чи опосередковано відмовою або відмовою іншого об'єкта. Причина поломки прихована під час проектування, виготовлення або використання об'єкта, що призвело до поломки. Механізм несправності може бути пов'язаний з фізичними, хімічними або іншими процесами, що призвели до несправності.

Наслідки відмови - явища, процеси, події, викликані відмовою об'єкта. Для розрахунку надійності використовуються такі терміни: Час роботи - це тривалість або обсяг роботи об'єкта, який може бути безперервним (час) або дискретним (кількість циклів) значенням.

1.3. Види показників надійності

Всі властивості надійності (безвідмовність, збереженість, ремонтпридатність) характеризуються показниками.

Показник надійності - це комплексна характеристика одного або декількох з тих властивостей, які в сукупності складають надійність об'єкта.

Серед показників надійності розрізняють:

- одінічні - це показники надійності, що характеризують одні з тих властивостей, які в сукупності складають надійність об'єкта;
- комплексні - це показники надійності, що характеризують кілька властивостей з тих, які в сукупності складають надійність об'єкта;
- експлуатаційні - це показники надійності, точкову або інтервальную оцінку якого визначають за результатами експлуатації;
- експериментальні - це показники надійності, точкову або інтервальную оцінку якого визначають за даними випробувань;
- розрахункові - визначають шляхом розрахунків. Кожна з властивостей надійності (безвідмовність, збереженість, ремонтпридатність) визначається набором показників.

До показників безвідмовності відносяться:

- ймовірність безвідмовної роботи;
- середній напрацювання до відмови;
- середній напрацювання між відмовами;
- γ - процентний напрацювання на відмову;
- інтенсивність відмов;
- середня інтенсивність відмов;
- параметр потоку відмов;
- середній параметр потоку відмов.

Імовірність безвідмовної роботи - це ймовірність того, що протягом заданого напрацювання відмова об'єкта не виникне (на початку інтервалу часу або напрацювання об'єкт не відмовить).

Середнє напрацювання до відмови - математичне очікування напрацювання об'єкта до першої відмови.

γ - відсотковий напрацювання на відмову - це напрацювання протягом якого відмова об'єкта не виникне з імовірністю γ вираженою у відсотках.

Інтенсивність відмов - умовна щільність ймовірності виникнення відмови об'єкта, що визначається за умови що до цього моменту відмова не виникла.

Параметр потоку відмов - відношення математичного очікування числа відмов відновлюваного об'єкта за досить малий напрацювання до значення цього напрацювання.

До показників довговічності відносяться:

- середній ресурс - це математичне очікування ресурсу;
- γ -відсотковий ресурс - це сумарна напрацювання протягом якого об'єкти не досягне граничного стану з імовірністю γ , ураженого в процентах; - середній термін служби - це математичне очікування терміну служби;
- γ -відсотковий термін служби - це календарна тривалість експлуатація об'єкта протягом якого він не досягне свого граничного стану з імовірністю γ вираженою у відсотках.

До показників зберігання слід віднести:

- середній термін зберігання - це математичне очікування терміну зберігання;
- γ -процентний термін зберігання - це термін зберігання що його досягає об'єкт із заданою вірогідністю γ , вираженою у відсотках.

До показників ремонтоздатності відносяться:

- ймовірність відновлення - ймовірність того, що час відновлення працездатного стану об'єкта не перевищує заданого значення; - середня тривалість відновлення;
- γ - процентна тривалість відновлення;
- інтенсивність відновлення - умовна щільність ймовірності відновлення працездатного об'єкта визначення для одного моменту часу за умови, що до цього моменту відновлення об'єкта не завершилося;
- середня інтенсивність відмов;

- середня трудомісткість технічного обслуговування (ремонту) - математичне очікування тих. обслуговування або ремонту виражене в людино-годин.

До комплексних показників надійності належить коефіцієнт готовності k_g - ймовірність того, що об'єкт виявиться працездатним в довільний момент часу, крім запланованих періодів, протягом якого використання об'єкта за призначенням не заплановано.

1.4. Класифікація Веб-сайтів

Основні види сайтів: соціальні мережі, бізнес сайти, веб-сервіс, інформаційні сайти, комбіновані сайти.

Соціальна мережа - спрямована на побудову спільнот в Інтернеті з людей зі схожими інтересами та / або діяльністю. Зв'язок здійснюється за допомогою сервісу внутрішньої пошти або миттєвого обміну повідомленнями.

Представницький сайт - так іноді називають сайт-візитку з розширеною функціональністю: докладний опис послуг, портфолію, відгуки, форма зворотнього зв'язку і т.д.

Корпоративний сайт - містить повну інформацію про компанії-власника, послуги / продукцію, події в житті компанії. Відрізняється від сайту-візитки і представницького сайту повнотою наданої інформації, часто містить різні функціональні інструменти для роботи з контентом (пошук і фільтри, календарі подій, фотогалереї, корпоративні блоги, форуми). Може бути інтегрований з внутрішніми інформаційними системами компанії-власника (KIC, CRM, бухгалтерськими системами). Може містити закриті розділи для тих чи інших груп користувачів - співробітників, дилерів, контрагентів тощо.

Комерційні сайти - до комерційних слід віднести ті сайти, які безпосередньо пов'язані з веденням бізнесу. З цієї точки зору серед них можна виділити: стимулюючі продажу в «офлайн» - бізнес (тобто бізнесі, який сам по собі існує

незалежно від існування Інтернету), орієнтовані на онлайн - комерцію (тобто ті види бізнесу, які без Інтернету неможливі, наприклад інтернет-магазини).

Сайт-візитка - складається буквально з п'яти-десяти сторінок і містить загальні відомості про власника сайту: його діяльність, реквізити, контактна інформація. Тобто такий сайт дійсно являє собою розширену візитну картку підприємця або організації.

Основною аудиторією комерційного сайту є дійсні і потенційні клієнти. За своїм обсягом і типом комерційний сайт може бути:

- Інтернет-магазин - містить каталог продукції, з якого користувач може прямо на сайті вибрати і замовити потрібні йому товари. Тобто обов'язковою ознакою інтернет-магазину є можливість зробити онлайн замовлення. Найчастіше ця можливість доповнюється функцією «кошика», яка дозволяє за один раз замовити кілька сподобалися товарів. сайт інфологіческая модель;
- Промо-сайт - створюється для презентації будь-якого конкретного товару або торгової марки і є частиною загальної маркетингової програми по його просуванню. Містить докладний опис технічних характеристик і переваг товару, інформацію про різні акції, знижки і т.п.;
- Бізнес інтернет-портал - зазвичай належать великим компаніям з широким діапазоном видів діяльності. Містить дуже велика кількість сторінок, охоплює велику кількість тем. Відмінною рисою є надання користувачу широкого діапазону різних сервісів: пошта, форум, чат і т.п.;
- Некомерційні сайти - зазвичай здійснюють надання користувачам різної інформації (інформаційні сайти) або надання будь-яких онлайн послуг (веб-сервіси);
- Особисті сторінки - так звані homepages (домашні сторінки), на яких людина розміщує інформацію про себе, фотографії та будь-яку ще інформацію по своєму бажанню. В даний час практично повністю витіснені блогами і соціальними мережами;

- Тематичні інформаційні сайти - присвячені будь-якої певної, досить вузької тематики. Як приклади можна навести численні сайти шанувальників певних письменників, фільмів, музичних груп;
- Пошукові системи. В даний час велика частина пошукових систем перетворилися на повноцінні інтернет-портали (Google, Яндекс, Рамблер). Проте як приклад «чистої» пошукової системи можна назвати Nigma.ru;
- Поштові сервіси - всі великі поштові сервіси або стали великими Інтернет - порталами (наприклад, Mail.ru), або є частиною інших порталів (наприклад, Gmail.com як сервіс, що входить в Google, або Яндекс.Пошта як частина Яндекса);
- Соціальні мережі - набули останнім часом просто фантастичну популярність. Концептуальний принцип таких веб-сайтів базуються на соціологічних теоріях соціальних мереж. Фактично їх функції полягають у можливості зареєстрованих користувачів знаходити в даній мережі певних людей і підтримувати з ними зв'язок, організовуватися в спільноти за інтересами, а також у наданні користувачам можливості розміщувати власний текстовий контент. Найбільшими соціальними мережами в світі є Facebook і MySpace, а в Росії - «Вконтакте» і «Однокласники»;
- Блогові сервіси - Надають майданчики для ведення блогів, дозволяють користувачам вести колективні блоги (спільноти) і організовувати зручне взаємне читання. Слово «блог» (blog) - це скорочення від web log - «веб-журнал». Фактично вдає із себе публічний онлайн щоденник. В даний час блоги мають дуже широкий спектр використання: від побутових заміток для кількох друзів, до політичної трибуни для тисяч людей. Як приклади блогових сервісів можна привести livejournal.com, liveInternet.ru, сервіс мікроблогів twitter.com та інші;
- Сервіси для зберігання фотографій - дані сервіси дають користувачам безкоштовний хостинг фотозображень і можливості для їх структурування та класифікації. Найбільш популярними фотохостинг є flickr.com, photobucket.com;

- Сервіси для зберігання і відтворення відео - на такі сайти користувачі можуть завантажувати свої відеозаписи, які можуть переглядати всі бажаючі. Більш того, сервіси надають можливість розмістити це відеозображення на будь-якому сайті. Найбільшим місцем для розміщення відеофайлів є YouTube.com, в Росії велику популярність отримав RuTube.ru;
- Дошки оголошень - фактично аналогічні звичайним побутовим дошкам оголошень або рекламним газетам. Кожен бажаючий може вивісити своє оголошення, а всі відвідувачі сайту - прочитати його. Дошки оголошень, як правило, поділені на кілька тематичних розділів;
- Каталоги сайтів - містять структуровані списки посилань на веб-сайти, забезпечені їх короткими описами. Масштаб різних каталогів може сильно різнитися: від декількох сотень посилань до глобальних всесвітніх рейтингів. До останніх слід віднести DMOZ і каталог Yahoo!. Російськомовні каталоги - Яндекс.Каталог, каталог mail.ru;
- Тематичні інформаційні Інтернет - портали представляють собою розширену версію тематичних сайтів з великою кількістю новин, статей та інших матеріалів. Зазвичай мають більш широкою тематикою, ніж тематичні сайти, і надають користувачам різні сервіси та інші можливості. Як приклад можна привести портали avto.ru, sports.ru;
- Онлайн енциклопедії - в переважній більшості випадків базується на вікі-принципі, який полягає в тому, що будь-який користувач може самостійно додавати нові статті та вносити зміни (редагування) в уже існуючі;

Очевидно, що більша частина перерахованих вище сайтів тим чи іншим способом займаються зароблянням грошей (розміщенням чи реклами, або будь-якими платними послугами для користувачів). Однак такі сайти впершу чергу створені не для отримання прибутку, а для задоволення потреб користувачів, тому їх можна назвати некомерційними. Втім, подібні розмежування вельми примарні, так що багато хто з цих сайтів можна віднести до категорії «умовно некомерційних».

1.6. HTML – мова розмітки гіпертексту

Для реалізації гіпертекстової технології була вибрана мова HTML. У програмі WWW-клієнт - це інтерпретатор HTML. Залежно від команди HTML-програми виконує різні функції: зміна тексту на екрані; обмін інформацією з сервером у міру аналізу отриманого HTML-тексту або графічних образів; вихід зовнішніх програм для роботи з документами у форматах, відмінних від HTML (GIF, JPEG, MPEG, Postscript та ін.).

Найбільш мобільного поширення набули веб-браузери Microsoft Internet Explorer і Netscape Navigator (Netscape Communicator).

Сервер у WWW працює за протоколом HTTP. Він працює за допомогою WWW, керує наявними базами даних у форматі HTML та програмним забезпеченням, розробленим у державі CGI. Отримали популярність таких серверів HTTP, як Netsite (Netscape), Apache (Apache Group), Open Market Secure Web Server (Open Market), NCSA HTTP (NCSA), Netscape Enterprise Server (Netscape), O'Reilly Website (O'Reilly), Інформаційний сервер Інтернету (Microsoft), Персональний веб-сервер (Microsoft) [1].

Сервіс WWW може організувати величезну кількість інформації в компактному сховищі з легким доступом до наявних матеріалів.

Найбільшою єдиною інформацією в сервісі W3 є веб-сторінка. Вона може мати різні розміри - від одного екрану до нескінченного їх кількості. Головна особливість веб-сторінок - використання нічого гіпертекстових посилань, активізація яких може переходити на інші частини веб-сторінок або на іншу сторінку.

Сукупність веб-сторінок, об'єднаних єдиною темою та організованих для спільної роботи, створює веб-сайт, або веб-представництво, або просто «веб».

Кожна веб-сторінка зберігається у файлах, ім'я якого має розширення «htm» (для MS DOS і Windows - платформа) або «html» (для Unix - платформа). З цього розширення знаходять веб-сторінки, програми-клієнти сервісу WWW-браузери

(перегляд веб-сторінок). Причому, браузери розуміють різні розширення сторінці, вони діляться на два види: пасивні та активні. Пасивна веб-сторінка містить стилі лише інформаційного наповнення (текст, графіка). При цьому для зберігання в графічній графіці EOM оформляється в окремих файлах (в основному - у форматі GIF, JPEG або PNG). Сама ж веб-сторінка (тобто файл із розширенням «htm») містить лише розміщений текст.

Активні веб-сторінки, крім вмісту, містять програми, за допомогою яких надходить інформація від відвідувачів (інтерактивна веб-сторінка), здійснюється зв'язок із СУБД (динамічно оновлювана веб-сторінка), банерна реклама відображається в просторі Веб-сторінка (Веб-сторінка з банерами), реалізовані деякі спеціальні ефекти, наприклад, за допомогою прокрутки, а інший сайт, крім інформаційних сторінок, повинен містити кореневу сторінку - домашню сторінку («домашня сторінка»). Він має стандартну назву в Інтернеті: index. html або index.htm. сайт, який, крім домашньої сторінки, не має інформаційних сторінок, а замість домашньої сторінки створюється лише її заглушка, називається «порожньою мережею».

Створення веб-сторінки полягає у розмітці тексту, що міститься на ній, підготовленому будь-яким текстовим редактором. Інші типи інформації (графіка, звук, анімація) готуються відповідними редакторами та форматуються як файли з відповідними розширеннями, які є стандартними для використовуваного браузера. Посилання між файлами, що використовуються на веб-сторінці, здійснюються за допомогою гіперпосилань.

Розмітка полягає у форматуванні тексту - визначенні його місця на сторінці та способу його відображення, вказуючи, куди на сторінці повинна бути вставлена графіка, як вона відноситься до тексту (вставляється у вільний простір, це фон, на якому текст застосовується, "обгортається" текстом тощо) та вставляється гіперпосилання, які переходять до інших веб-сторінок або програм, що використовуються. Для розмітки використовується HTML.

1.7. CSS – каскадні таблиці стилей

Під терміном CSS ми маємо на увазі фразу "Каскадні таблиці стилів". Вони вбудовані в HTML-код і не потребують спеціальних редакторів та компіляторів, оскільки інтерпретуються звичайним браузером.

Каскадні таблиці стилів - це описи різних елементів HTML і призначені для розширення властивостей останніх. Вперше стилі були запропоновані WWW Consortium як частина розробки специфікації HTML 3.0, але насправді ця шняга почала підтримуватися лише в 1997 році. Тільки щасливі власники браузерів, таких як Netscape Navigator 4.0 та Internet Explorer 4.0 мали можливість користуватися CSS. Він використовується творцями веб-сторінок для встановлення кольорів, шрифтів, макета та інших аспектів презентації документа.

Основною метою розробки CSS було розділення вмісту (написаного HTML або іншою мовою розмітки) та презентації документа (написаного CSS). Цей поділ може збільшити доступність документа, забезпечити більшу гнучкість та контроль над його поданням, а також зменшити складність та повторюваність структурованого вмісту. Крім того, CSS дозволяє відображати один і той самий документ у різних стилях або методах виводу, таких як відображення, друк, читання голосом (із виділеним голосовим браузером чи програмою зчитування з екрана) або виведення з пристроїв, що використовують шрифт Брайля [2].

Стандарт CSS визначає пріоритети, в послідовності застосування правил стилю, якщо властивості кількох правил підходять для елемента одночасно (або, у рідкісних випадках, в одному правилі є властивості з тією ж назвою). Це називається "каскадом", в якому для властивостей розраховуються пріоритети або "ваги", щоб зробити результати передбачуваними.

1.8. Bootstrap – інструмент з шаблонами CSS та HTML

Давно залишилися в минулому часи, коли створення сайтів було прерогативою виключно професійних дизайнерів, прочитали купу розумних книжок і мають диплом про відповідну освіту. Колись для вирішення цього завдання потрібні були спеціальні знання, незвичайний практичний досвід і великий запас власних напрацювань.

В принципі, сьогодні всі ці якості також не для веб-дизайнера зайвими, але при наявності бажання будь-який початківець, який прийшов з якоїсь суміжній галузі інтернет-комерції, має можливість зверстати досить пристойний сайт, який буде відповідати всім стандартним вимогам і запитам користувачів. Адже тепер на допомогу такому хороброму є цілий набір спеціальних інструментів, використання яких спрощує завдання створення стандартизованих інтерфейсів.

Однією з таких програм є Bootstrap. Ця розробка є прекрасним прикладом простого у виконанні фреймворка, за допомогою якого практично будь-який зацікавлений та зацікавлений користувач може самостійно створити сучасний інтерфейс веб-переглядача.

Серед основних переваг програми фахівці в галузі веб-дизайну відзначають ретельно продуману структуру кодування HTML, а також органічну адаптацію до оболонки програм JavaScript і CSS. Все це відкриває перед користувачем широкі можливості для створення веб-сайтів з використанням різноманітних елементів. Популярність Bootstrap постійно зростає не тільки серед «манекенів» веб-дизайну, дедалі більше випадків, коли професіонали вдаються до використання програми. Це спонукає творців постійно розробляти ці рамки.

Що входить в Bootstrap:

- Сітки - це попередньо розмірені стовпці, щоб їх можна було негайно використовувати для створення окремих елементів, наприклад, при описі документа за допомогою CSS. Таким чином, за допомогою Bootstrap ви можете швидко і досить легко створити каркас сторінки, встановити дизайн знизу і вгорі, кількість і ширину стовпців, в яких потім можна розмістити інші компоненти сайту;

- Шаблони документів - у перших версіях кількість готових шаблонів була дуже обмежена, але з розвитком програми цей недолік був усунутий. Сьогодні бібліотека - це досить багата колекція шаблонів документів, в якій ви можете знайти як фіксовані, так і «текучі» зразки;
- Типографіка - цей інструмент передбачає заздалегідь створені описи шрифтів, включаючи використання класів, таких як код, цитата тощо;
- Медіа - широкий спектр опцій для управління статичними зображеннями та відеофайлами;
- Таблиці - за допомогою цього інструменту ви можете не тільки створювати повноцінні таблиці зі складною структурою, але і додавати для них функціональність, наприклад, ви можете надати можливість сортування даних, що містяться в таблиці, на основі різних характеристик;
- Форми - класи, за допомогою яких ви можете не тільки створювати та проектувати різні форми, але й події, що відбуваються з ними;
- Сповіщення - інструменти для проектування діалогових вікон, підказок тощо.

1.10. Постановка задачі

Метою роботи є розробка веб-сайту «Simple» Online ресторан на основі WEB – технології, що надає можливість підвищити ефективність людських ресурсів і покращити комунікацію.

Задачами, які мають виконуватись даним продуктом є:

- створення серверної частини;
- розробка веб-сайту.

Система має забезпечувати наступні можливості:

- переглядати інформацію про ресторан, його контактні данні, місце знаходження, години роботи;

- переглядати опис самого ресторану;
- переглядати меню;
- бронювати стіл;
- переглядати галерею;
- читати відгуки;
- підписуватись, щоб отримувати ексклюзивні рецепти від шеф-кухаря.

Висновки до розділу 1

В результаті виконаної роботи було проаналізовано теоретично частину на постановку задачі. Було розглянуто відмови, помилки, види несправності, види показників надійності, параметр потоку відмов. Також саме поняття веб-сайту, його класифікацію. Також проаналізовано основну будову сайту, HTML – мова розмітки гіпертексту, каскадні таблиці CSS, бібліотеку Bootstrap.

РОЗДІЛ 2

ЗАСОБИ РЕАЛІЗАЦІЇ ТА РОЗРОБКИ

2.1. Пакетний менеджер npm

Крім вбудованих і кастомних модулів Node.js існує величезний пласт різних бібліотек і фреймворків, різноманітних утиліт, які створюються сторонніми виробниками і які також можна використовувати в проекті, наприклад, express, grunt, gulp і так далі. І вони теж нам доступні в рамках Node.js. Щоб зручніше було працювати з усіма сторонніми рішеннями, вони поширюються у вигляді пакетів. Пакет по суті являє набір функціональностей.

Для автоматизації установки і оновлення пакетів, як правило, застосовуються систему управління пакетами або менеджери. Безпосередньо в Node.js для цієї мети використовується пакетний менеджер NPM (Node Package Manager). NPM за замовчуванням встановлюється разом з Node.js, тому нічого доустановлювати не потрібно. Але можна оновити встановлену версію до самої останньої.

NPM також може бути використаний для (глобального) встановлення Express Application Generator, зручного інструменту для створення скелетних веб-програм Express, які відповідають шаблону MVC. Генератор додатків необов'язковий, оскільки не потрібно використовувати цей інструмент для створення програм, які використовують Express, або для створення додатків Express, які мають однаковий архітектурний макет або залежності.

Кафедра КІТ (47)				НАУ 21 42 93 000 ПЗ							
Виконав	Сохань Ю.С.			ЗАСОБИ РЕАЛІЗАЦІЇ ТА РОЗРОБКИ	Літера		Аркуш		Аркушів		
Керівник	Климова А.С.							23	15		
Консульт.											
Н-котрол.	Шевченко О.П.										
Зав. каф.	Савченко А.С.										
					412 122						

2.2. Мова програмування JavaScript

JavaScript - інтерпретована мова програмування, що є однією з реалізацій мови ECMAScript і заснований на концепції прототипів, що прийшла з мови Self.

В даний час JavaScript використовується в основному для створення сценаріїв поведінки браузера, що вбудовуються в веб-сторінки, але також знаходить застосування в якості скриптової мови доступу до об'єктів додатків. Деякі додатки мають вбудований інтерпретатор JavaScript, що дозволяє розширювати їх можливості без зміни самого додатка.

Мова програмування JavaScript розроблений фірмою Netscape у співпраці з Sun Microsystems і призначений для створення інтерактивних HTML-документів. Основні галузі використання JavaScript такі:

- створення динамічних сторінок, тобто сторінок, вміст яких може змінюватися після завантаження документа;
- перевірка правильності заповнення користувачем форм до пересилання їх на сервер;
- рішення "локальних" завдань за допомогою сценаріїв і деякі інші сфери, дозволяють створювати додатки, які виконуються як на стороні клієнта, так і на стороні сервера. При розробці додатків обох типів використовується так зване ядро, в якому містяться визначення стандартних об'єктів. Клієнтські програми виконуються браузером на машині користувача. Програма (сценарій) на мові JavaScript обробляється вбудованим в браузер інтерпретатором. Треба прагнути до того, щоб написані сценарії коректно виконувалися у будь-якому браузері.

Код JavaScript включається в HTML-код сторінки і виконується інтерпретатором, вбудованим в браузер.

2.2.1. JavaScript на стороні клієнта

На стороні клієнта JavaScript є найпоширенішою формою мови. Сценарій повинен бути включений в HTML документ або на нього посилатися, щоб код був інтерпретований браузером. Код JavaScript виконується, коли користувач подає форму, і тільки якщо всі записи дійсні, вони будуть подані на веб-сервер. JavaScript може використовуватися для уловлювання ініційованих користувачем подій, таких як кліки кнопок, навігація посилань та інші дії, які користувач ініціює явно або неявно.

Механізм на стороні клієнта JavaScript надає багато переваг перед традиційними сценаріями на стороні сервера CGI [3]. Наприклад, ви можете скористатися JavaScript, щоб перевірити, чи введе користувач дійсну адресу електронної пошти у полі форми. Тому веб-сторінка не повинна бути статичним HTML, але може включати програми, які взаємодіють з користувачем, керують браузером і динамічно створюють вміст HTML.

2.2.2. Переваги JavaScript

Переваги використання JavaScript:

- Жоден сучасний браузер не обходиться без підтримки JavaScript;
- Використання плагінів та сценаріїв, написаних на JavaScript, не може зробити навіть фахівець;
- Корисні функціональні налаштування;
- Постійно розвивається мова - це бета-версія проекту, JavaScript2;
- Взаємодія з додатком може здійснюватися навіть за допомогою текстових редакторів - Microsoft Office та Open Office;
- Перспективи використання мови в процесі викладання програмування та інформатики;

- Більш чудові інтерфейси — ви можете використовувати JavaScript, щоб включити такі елементи, як компоненти перетягування і повзунки, щоб надати відвідувачам свого сайту багатий інтерфейс;
- Незамінність для веб-розробки. Підтримка скриптів усіма популярними браузерами; повна інтеграція з версткою сторінок (HTML + CSS) і серверної частиною (backend);
- Швидкість роботи і продуктивність. Javascript дозволяє частково обробляти веб-сторінки на комп'ютерах користувача без запитів до сервера. Це економить час і трафік, знижує навантаження на сервер;
- Потужна інфраструктура (екосистема). Перші 10 років цього не було і в помині. Потім кількість готових рішень в відкритому доступі так зросла, що працювати з Javascript і його фреймворками стало задоволенням;
- Простота і раціональність застосування. Просту завдання можна вирішити за 5 хвилин, не треба робити зайву роботу. Для складних завдань є варіанти вирішення, можна підібрати кращий, адаптувати;
- Зручність для користувача інтерфейсів. Заповнення форм, вибір дій, активація кнопок, перевірки введення, реагування на наведення / кліки миші і т.п Це дає приголомшливий рівень юзабіліті;

2.2.3. Обмеження JavaScript

Ми не можемо розглядати JavaScript як повноцінну мову програмування. Вона не має таких важливих функцій:

- Знижений рівень безпеки через повсюдного і вільного доступу до вихідного коду популярних скриптів;
- Безліч дрібних дратівливих помилок на кожному етапі роботи. Велика частина з них легко виправляється, але їх наявність дозволяє вважати цю мову менш професійним, порівняно з іншими;

- Підтримка браузера: JavaScript іноді інтерпретується по-різному різними браузерами. Різні механізми компонування можуть відображати JavaScript по-різному, що призводить до неузгодженості з точки зору функціональності та інтерфейсу. Велика частина JavaScript залежить від маніпуляції елементами DOM браузерів. І різні браузери надають різні типи доступу до об'єктів, зокрема Internet Explorer;
- Завантаження файлу: файл JavaScript завантажується на клієнтській машині, щоб кожен міг прочитати код і повторно використовувати його;
- Відключити JavaScript: якщо ви вимкніть JavaScript в браузері, весь код JavaScript не запущено;
- JavaScript не можна використовувати для мережесих програм, оскільки такої підтримки немає [4].

2.3. Бібліотека реалізації веб-інтерфейсу React.js

React JS - це бібліотека JavaScript, яка використовується у веб-розробці для створення інтерактивних елементів на веб-сайтах. Але якщо ви не знайомі з JavaScript або бібліотеками JavaScript, це не корисне визначення. Тож давайте зробимо крок назад і спершу розберемося з цими умовами [5].

React - це бібліотека JavaScript, яка спеціалізується на допомозі розробникам у створенні користувацьких інтерфейсів або інтерфейсів користувача. Що стосується веб-сайтів та веб-додатків, інтерфейси користувача - це набір екранних меню, рядків пошуку, кнопок та всього іншого, з чим хтось взаємодіє для використання веб-сайту чи програми.

До React JS розробники застрягли у створенні користувацьких інтерфейсів вручну за допомогою «ванільного JavaScript» (розробник самостійно говорить за необроблену мову JavaScript) або з менш орієнтованими на UI попередниками React, такими як jQuery. Це означало довший час розробки та безліч можливостей для помилок та помилок.

Окрім забезпечення багаторазового коду бібліотеки React (економія часу на розробку та зменшення шансів на помилки кодування), React має дві ключові функції, які додають привабливості розробникам JavaScript:

- JSX;
- Віртуальний DOM.

Різниця між бібліотеками JavaScript (як React) та фреймворками JavaScript (як Angular полягає в тому, що - у випадку з бібліотекою - розробник застосовує бібліотечний код в окремих екземплярах, які вимагають цього. Однак, коли мова заходить про фреймворки, фреймворк створює риштування, що упорядковує ваш веб-сайт або додаток та забезпечує виділені області для підключення коду фреймворку.

React JS іноді приймають за повномасштабний фреймворк, оскільки його надійна екосистема та розширюваність роблять її такою універсальною бібліотекою JavaScript. Пам'ятайте, коли ви використовуєте React JS для створення інтерфейсів веб-сайтів та веб-додатків, ви маєте доступ до:

- Фрагмент коду та компоненти React (будівельні блоки коду React, що використовуються для створення певних частин користувацького інтерфейсу);
- Можливість використання JSX для безпосереднього маніпулювання DOM Віртуальний DOM для покращення продуктивності вашого веб-сайту.

Але крім усього цього, React JS - це проект з відкритим кодом, тобто кожен може безкоштовно завантажити та змінити його вихідний код. Це також означає, що незалежно від конкретної функції інтерфейсу, яку ви сподіваєтеся реалізувати з React JS, існує бібліотека React, яка відповідає вашим потребам. Розмір вашої бібліотеки React може зростати в геометричній прогресії за допомогою доповнених бібліотекою React спільних бібліотек, починаючи від колекцій окремих функцій інтерфейсу і закінчуючи шаблонами React JS для побудови інтерфейсу з нуля.

2.4. Віртуальний DOM

DOM розшифровується як "Модель об'єкта документа". DOM простими словами представляє інтерфейс вашої програми. Щоразу, коли змінюється стан інтерфейсу вашого додатка, DOM оновлюється, щоб відобразити цю зміну. Зараз улов часто маніпулює DOM, впливає на продуктивність, роблячи її повільною.

DOM представлений у вигляді деревної структури даних. Через це зміни та оновлення DOM відбуваються швидко. Але після зміни оновлений елемент і його діти повинні бути повторно відтворені, щоб оновити інтерфейс програми. Повторне відображення або перефарбовування інтерфейсу - це те, що робить його повільним.

Віртуальний DOM - це лише віртуальне представлення DOM. Щоразу, коли змінюється стан нашого додатка, віртуальний DOM оновлюється замість реального DOM.

2.5. Single Page Application

Основна різниця між багатосторінковими та односторінковими програмами полягає в тому, що односторінкові програми завантажують сторінку один раз після її запуску.

Як тільки користувач взаємодіє з програмою, буде змінено лише необхідний компонент, а не повну програму, що робить односторінкову програму набагато швидшою з точки зору інтерактивності. Односторінкові програми пропонують набагато кращу взаємодію з користувачем (UX), тобто користувачі можуть легко переходити між різними сторінками програми, не чекаючи завантаження сторінок.

Single Page Application(SPA) - це веб-додаток або веб-сайт, який використовує один документ HTML як обгортку для всіх веб-сторінок і створює динамічно завантажувані HTML, CSS, JavaScript, як правило, взаємодію користувача за допомогою AJAX.

Операційні системи схожі на локальні (локальні) програми, з тією лише різницею, що вони працюють у браузері, а не в локальному процесі операційної системи.

2.5.1. Основні елементи SPA

Основні елементи у використанні Single Page Application:

- Кадри JavaScript, особливо MVC та MVVM;
- Напрямок: Навігація між видами здійснюється спереду;
- Шаблонний механізм;
- HTML5;
- API для бекенда, наприклад REST;
- Ajax.

2.5.2. Переваги і недоліки SPA

Головною перевагою односторінкових додатків є його швидкість. Більшість потреб SPA (HTML + CSS + Scripts) завантажуються під час запуску програми, і їх не потрібно перезавантажувати під час використання. Єдине, що змінюється - це дані, які передаються на сервер і з нього. Як результат, програма дуже чуйно реагує на запити користувача і не повинна весь час чекати на зв'язок клієнт-сервер.

З точки зору розробника, створення такого додатка впорядковується та оптимізується. Вам не потрібно писати код, щоб відображати сторінки на сервері. Вам навіть не потрібен сервер, щоб розпочати процес розробки. Ви можете розпочати роботу з файлу, щоб розпочати. Крім того, розробник може повністю використовувати той самий серверний код серверної частини та ефективний API для веб-програми та власного мобільного додатка.

Односторінкові додатки чудові, коли у вас є команда розробників, які працюють разом. Це дозволяє розробникам бекендів зосередитись на API, тоді як розробники інтерфейсу можуть приділяти більше уваги створенню найкращого досвіду для користувачів на основі інтерфейсу API та реалізації чудового інтерфейсу користувача.

Налагодження односторінкової програми також є простим за допомогою браузера Chrome, оскільки вона має спеціальні інструменти для Angular Batarang і React (технології, що використовуються для SPA). За допомогою консолі ви можете контролювати мережеві операції, а також досліджувати різні елементи сторінки та пов'язані дані.

Процес кешування також є досить ефективним - додаток надсилає лише один запит, зберігає всі передані дані і може використовувати ці дані. Це особливо важливо в періоди, коли користувач може мати поганий зв'язок - він / вона все ще може використовувати ваш додаток, оскільки він синхронізується із сервером, коли з'єднання покращується [6].

Незважаючи на всі переваги односторінкових програм, ніщо не буває ідеальним.

Односторінкові програми роблять більш значне навантаження на браузер. Наприклад, якщо ваші користувачі мають пристрої з низьким енергоспоживанням, вони матимуть поганий досвід роботи з швидкодією.

Додаткові проблеми з JavaScript виникають, тому що ви повинні переконатися у відсутності витоків пам'яті. Оскільки додаток може працювати тривалий час - годин за раз (на відміну від МРА, де тривалість сторінки рахувалася у хвиликах), вам потрібно переконатися, що ваш SPA не споживає більше пам'яті, ніж йому потрібно. В іншому випадку задоволення від швидко завантажуваних сторінок буде знищено повільністю недоступної пам'яті на пристрої користувача.

Ще одним недоліком JavaScript є те, що користувачі можуть просто вимкнути його на своїх пристроях, і тоді вам потрібно продумати додаткові способи, як вони можуть отримати доступ до інформації на вашому веб-сайті або у вашому додатку без JavaScript. Ще одна важлива річ, яку слід пам'ятати щодо використання або побудови SPA - це безпека. Завдяки міжсайтовому сценарію (XSS) та тому, що не завантажуються нові сторінки, хакери можуть отримати доступ до вашого веб-сайту та вводити нові сценарії на стороні клієнта.

Також односторінкові програми використовують JavaScript, тому виявити помилки може бути важко. Користувачам потрібно ввімкнути JavaScript у своєму

веб-браузері, інакше програма не буде працювати. Витік пам'яті може призвести до зниження продуктивності. Односторінкові програми не підходять для корпоративних програм, лише для малих та середніх програм. Вони мають жакливу ремонтпридатність завдяки використанню сторонніх плагінів.

2.5.3. Коли використовувати SPA

Якщо є бізнес або особистий веб-сайт, який потребує динамічної платформи та невеликого обсягу даних, хороша ідея - односторінковий додаток. Це також чудовий варіант, якщо ви плануєте розробляти мобільний додаток у майбутньому, оскільки, як ми вже згадували вище, серверний API можна використовувати як для Інтернету, так і для мобільних додатків.

Основним недоліком є SEO, але архітектура підходить для платформ Software-as-a-Service (SaaS), закритих спільнот та соціальних мереж (саме тому Facebook це використовує). Причиною цього є те, що ці сайти не потребують оптимізації для пошуку в Google.

2.6. Створення веб-сайту за допомогою React.js

Для забезпечення зручності роботи з даними було розроблено веб-сайту, написаний за допомогою мови програмування JavaScript.

Для створення проекту потрібно встановити бібліотеку create-react-app. Вона полегшує створення коду додатків і бібліотеки,

Для створення нового проекту потрібно виконати команду create-react-app project_name. Команда “yarn dev” запускає сервер, стежить за файлами і перебудовує додаток, коли ви зробите зміни в цих файлах. Параметр --open автоматично відкриває ваш браузер за адресою <http://localhost:3000/>.

Основною фундаментальною одиницею React є компоненти, які використовуються в якості будівельних блоків. В папці components знаходяться основні компоненти застосунку, в папці configs — налаштування, components —

сторінки застосунку, store — управління станом системи, styles — основні стилі, utils — допоміжні функції, assets — зображення. Представлена файлова структура веб-клієнта (рис. 2.1).

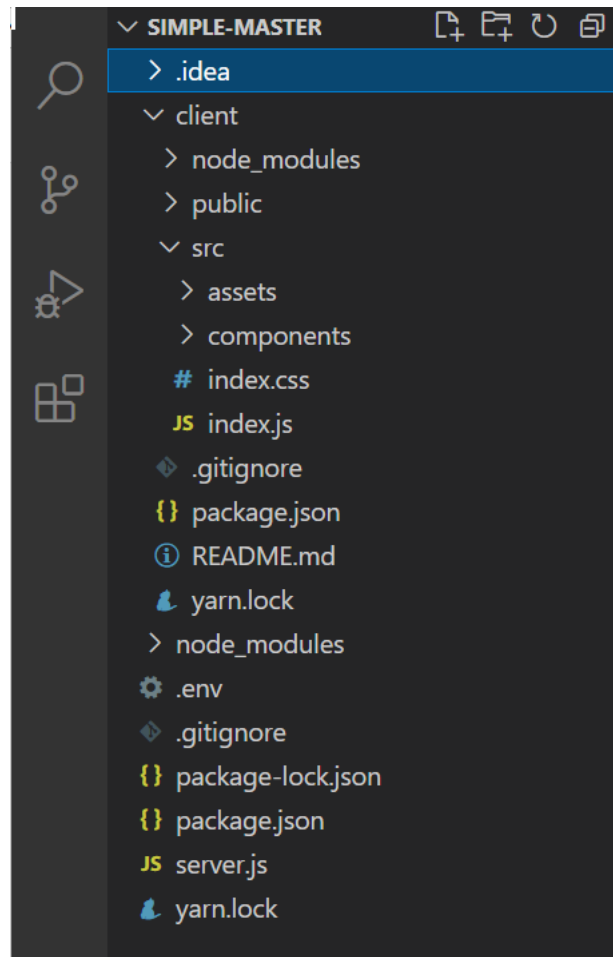


Рис. 2.1. Файлова структура веб-клієнта

2.7. Створення серверної частини з Express.js

Сервер було побудовано за допомогою платформи Node.js. Ця платформа дозволяє за допомогою мови програмування Javascript будувати rest api сервіси. З точки зору розробки веб-сервера Node має ряд переваг.

Чудова продуктивність. Node був розроблений для оптимізації пропускної здатності та масштабованості веб-додатків і є хорошим рішенням для багатьох загальних проблем веб-розробки (наприклад, веб-додатків у режимі реального часу).

Код написаний "простим старим JavaScript", що означає, що менше часу витрачається на "зміщення контексту" між мовами, коли ви пишете як на стороні клієнта, так і на стороні сервера [7].

JavaScript є відносно новою мовою програмування і отримує вигоди від вдосконалення мовного дизайну в порівнянні з іншими традиційними мовами веб-серверів (наприклад, Python, PHP тощо). Багато інших нових і популярних мов компілюються / перетворюються в JavaScript, щоб ви також могли використовувати TypeScript, CoffeeScript, ClojureScript, Scala, LiveScript тощо.

Менеджер пакетів вузлів (NPM) забезпечує доступ до сотень тисяч багаторазових пакетів. Він також має найкращу у своєму класі роздільну здатність залежностей, а також може бути використаний для автоматизації більшості ланцюжків інструментів побудови.

Node.js портативний. Він доступний в Microsoft Windows, macOS, Linux, Solaris, FreeBSD, OpenBSD, WebOS та NonStop OS. Крім того, він добре підтримується багатьма провайдерами веб-хостингу, які часто надають певну інфраструктуру та документацію для розміщення сайтів Node.

У ньому є дуже активна стороння екосистема та спільнота розробників, де багато людей готові допомогти.

Можна використовувати Node.js для створення простого веб-сервера за допомогою пакета Node HTTP.

Express - це мінімальна та гнучка структура веб-додатків Node.js, яка забезпечує надійний набір функцій для розробки веб- і мобільних додатків. Це сприяє швидкому розвитку веб-додатків на основі Node. Нижче наведено деякі основні особливості фреймворку Express:

- Дозволяє налаштувати проміжні програми для відповіді на запити HTTP;
- Визначає таблицю маршрутизації, яка використовується для виконання різних дій на основі методу HTTP та URL-адреси;
- Дозволяє динамічно відображати HTML-сторінки на основі передачі аргументів у шаблони.

Хоча сам Express є досить мінімалістичним, розробники створили сумісні проміжні пакети для вирішення практично будь-якої проблеми веб-розробки. Існують бібліотеки для роботи з файлами cookie, сеансами, логінами користувачів, параметрами URL, даними POST, заголовками безпеки та багатьма іншими. Ви можете знайти список пакетів проміжного програмного забезпечення, який підтримується командою Express у Express Middleware (разом із переліком деяких популярних сторонніх пакетів) [8].

Для створення проекту потрібно встановити express, створити екземпляр сайту і вказати порт на якому буде запущений сайт.

Для запуску сервера необхідно виконати команду `nodemon app.js`. Сервер виконує такі функції:

- виконує завантаження файлів;
- виконує відправку електронних листів;
- за допомогою http запитів реалізує зв'язок з клієнтською частиною.

Файл `package.json` містить список підключених модулів для даного проекту з `npm`, команди для запуску проекту. Список модулів з `package.json`:

```
body-parser: ^1.19.0",
concurrently: ^6.1.0",
cors: ^2.8.5",
dotenv: ^10.0.0",
express: ^4.17.1",
express-favicon: ^2.0.1",
nodemailer: ^6.6.0",
path: ^0.12.7".
```

2.8. Середовище розробки Visual Studio Code

Visual Studio Code (рис. 2.2) - це легкий, але потужний редактор вихідного коду, який працює на вашому робочому столі і доступний для Windows, macOS і

Linux. Він поставляється з вбудованою підтримкою JavaScript, TypeScript і Node.js і має багату екосистему розширень для інших мов (таких як C ++, C #, Java, Python, PHP, Go) і середовища виконання (наприклад, .NET і Unity).

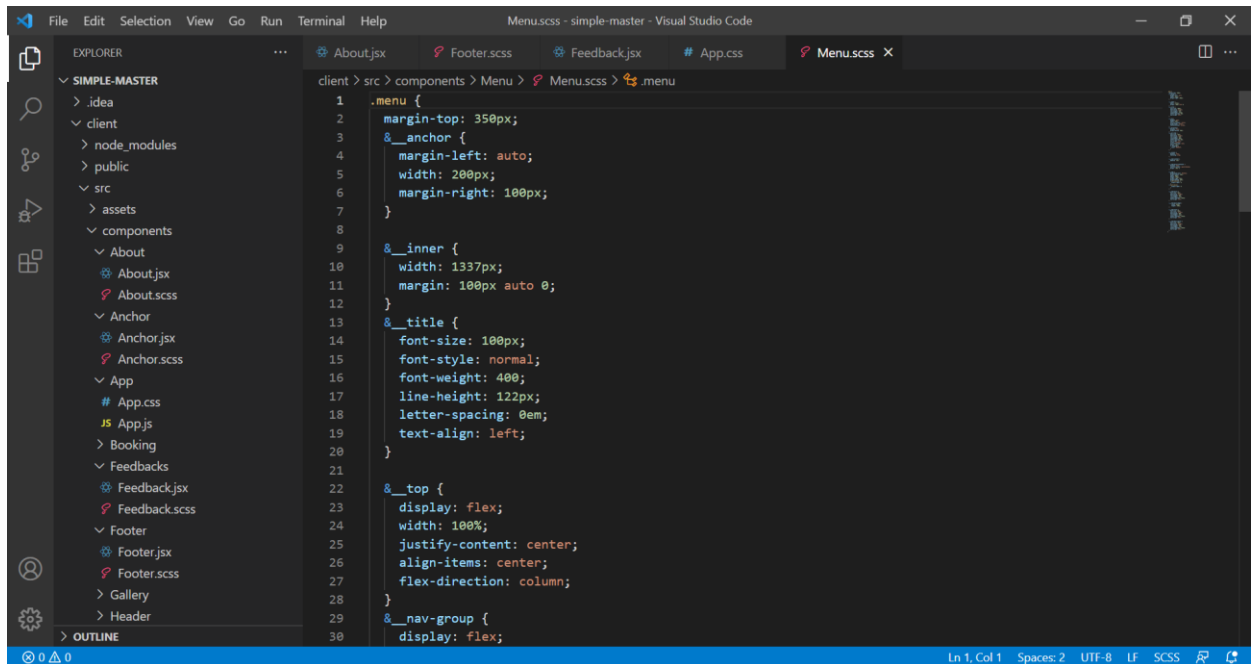


Рис. 2.2. Зразок інтерфейсу сайту середовища розробки Visual Studio Code

У Visual Studio Code у нас є підтримка майже всіх основних мов програмування. Деякі з них поставляються в комплекті, наприклад, JavaScript, TypeScript, CSS і HTML, але багатші мовні розширення можна знайти в VS Code Marketplace.

2.8.1. Особливості мови в VS Code.

Багатство підтримки розрізняється для різних мов і їх розширень:

- Підсвічування синтаксису і зіставлення дужок;
- Розумне завершення (IntelliSense);
- Літінг і виправлення;
- Навігація по коду (перейти до визначення, знайти всі посилання)
- Налаштування;

- Рефакторинг;
- Зміна мови для обраного файлу;

Висновки до розділу 2

В результаті виконаної роботи було пораналізовано бібліотека реалізації веб-інтерфейсу React.js. Чому слід використовувати ReactJS, його простоту, компонентний підхід, чітко визначений життєвий цикл і використання простого JavaScript, спеціальний синтаксис. Нативний підхід та легкість навчання бібліотеки. Прив'язка даних та продуктивність. Було проаналізовано Віртуальний DOM, його проблематика., Single Page Application, їх переваги та недоліки, та приклади застосування.

Також було проаналізовано та розроблено серверну частину за допомогою Express.js .

РОЗДІЛ 3

ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

Для використання реалізованого веб-застосунку необхідно мати встановленим будь-який Інтернет браузер. Потрібно переконатися, що ваша система відповідає даним вимогам.

Після входу на сайт, користувач має змогу переглядати інформацію про ресторан, його контактні данні, місце знаходження, години роботи, опис самого ресторану, переглядати меню, бронювати стіл, також переглядати галерею, читати відгуки та підписуватись, щоб отримувати ексклюзивні рецепти від шеф-кухаря (рис. 3.1)



Рис. 3.1. Головна сторінка

Кафедра КІТ (47)				НАУ 21 42 93 000 ПЗ			
Виконав	Сохань Ю.С.			ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ	Літера	Аркуш	Аркушів
Керівник	Климова А.С.					38	10
Консульт.					412 122		
Н-котрол.	Шевченко О.П.						
Зав. каф.	Савченко А.С.						

В цьому розділі можна побачити інформацію про ресторан «Simple» (рис. 3.2).



Рис. 3.2. Розділ о нас

На сайті, у розділі меню, можна переглядати страви, які подаються у ресторані, їх ціни, оцінка якості, а також з яких інгредієнтів вони складаються (рис. 3.3).

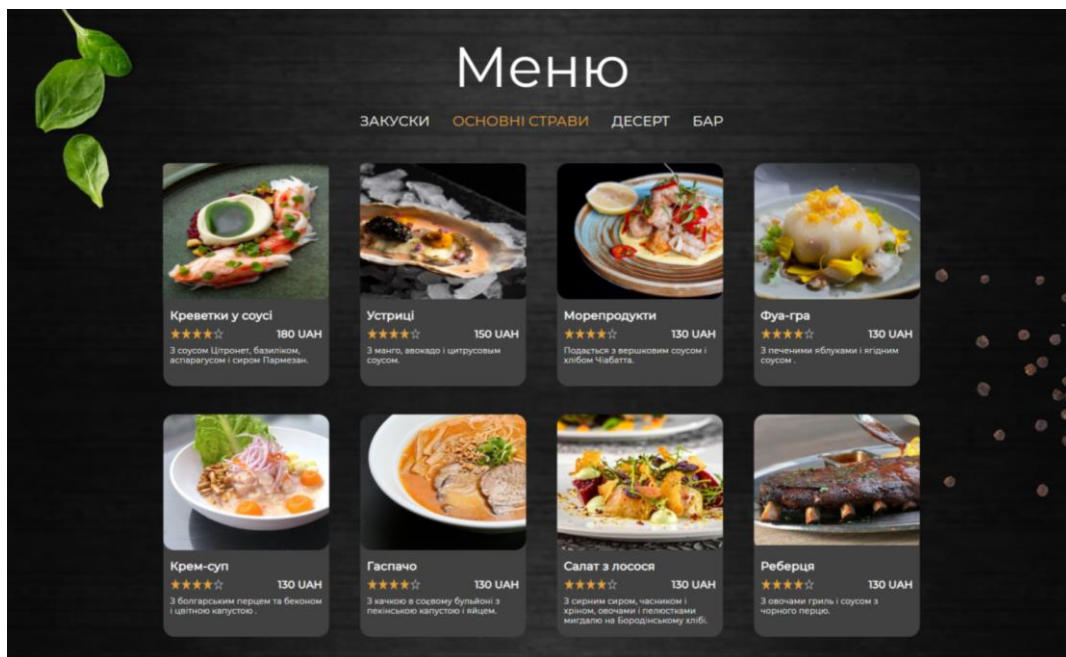


Рис. 3.3. Меню страв

У меню є навігація по розділам, яка підсвічується іншим кольором при виборі. Це закуски, основні страви, десерт та бар (рис. 3.4).



Рис. 3.4. Розділи страв в меню

Користувач має можливість забронювати стіл у ресторані онлайн. У формі він може ввести свої данні та будь-який коментар, якщо потрібен. У всіх полях форм, є маски-підказки для зручного користування (рис. 3.5).

Рис. 3.5. Форма для бронювання стола

А також має можливість обрати кількість гостей, час та дату, на який хоче замовити стіл (рис. 3.6).

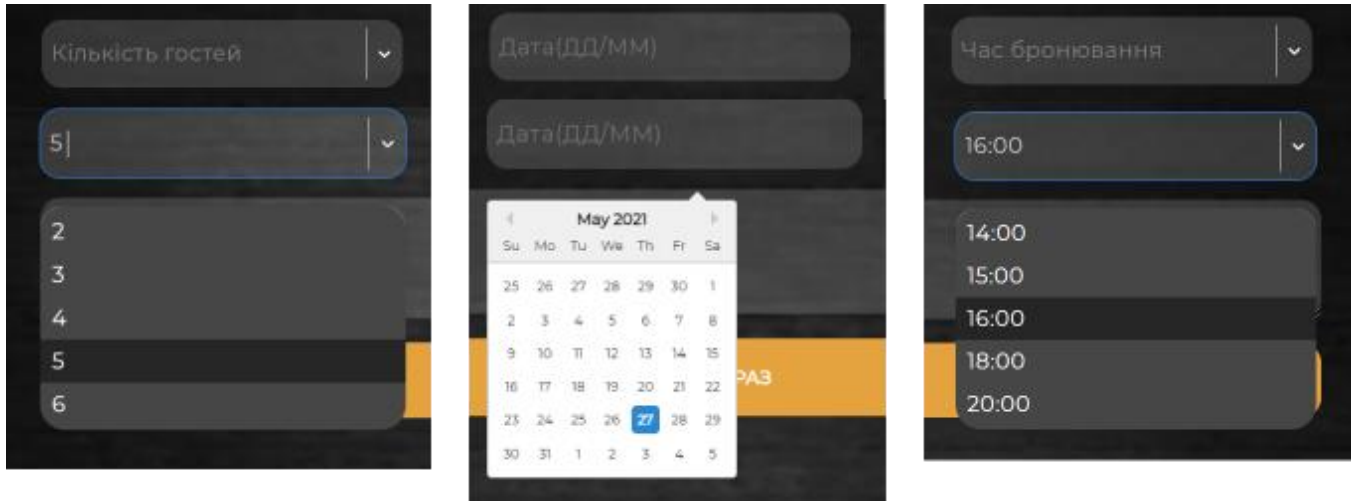


Рис. 3.6. Модальні вікна вибору кількості гостей, дати та часу

Після того як замолення зроблено, користувачу впливає повідомлення про успішне бронювання (рис. 3.7).

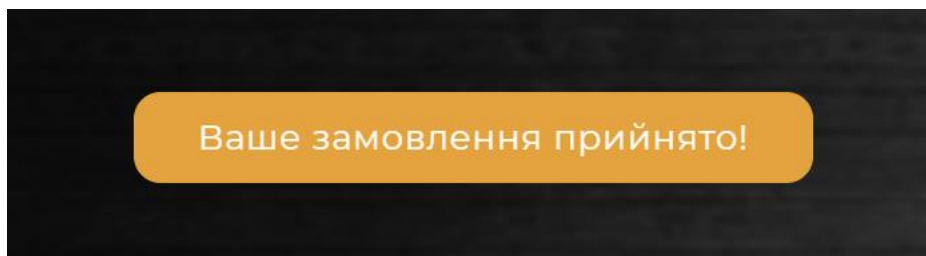


Рис. 3.7. Модальне вікно підтвердження бронювання столу

В цей час, адміну прийде лист на пошту з даними користувача, які він заповнив (рис. 3.8).

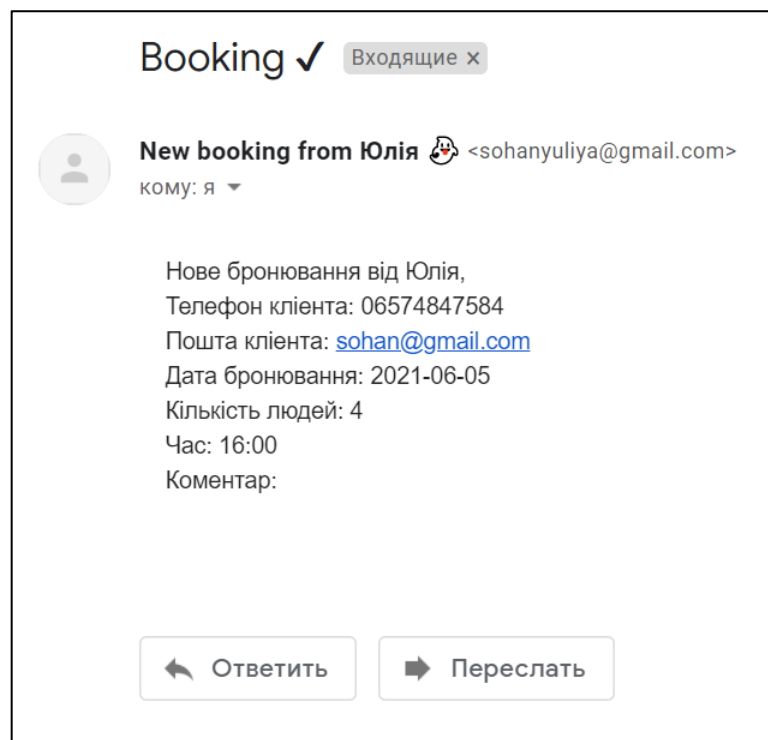


Рис. 3.8. Лист з даними користувача

Користувач має можливість переглянути галерею, в якій зібрані фото інтер'єру, екстер'єру, страв ресторану. Є також навігаційна панель для зручного переходу між розділами (рис. 3.9).

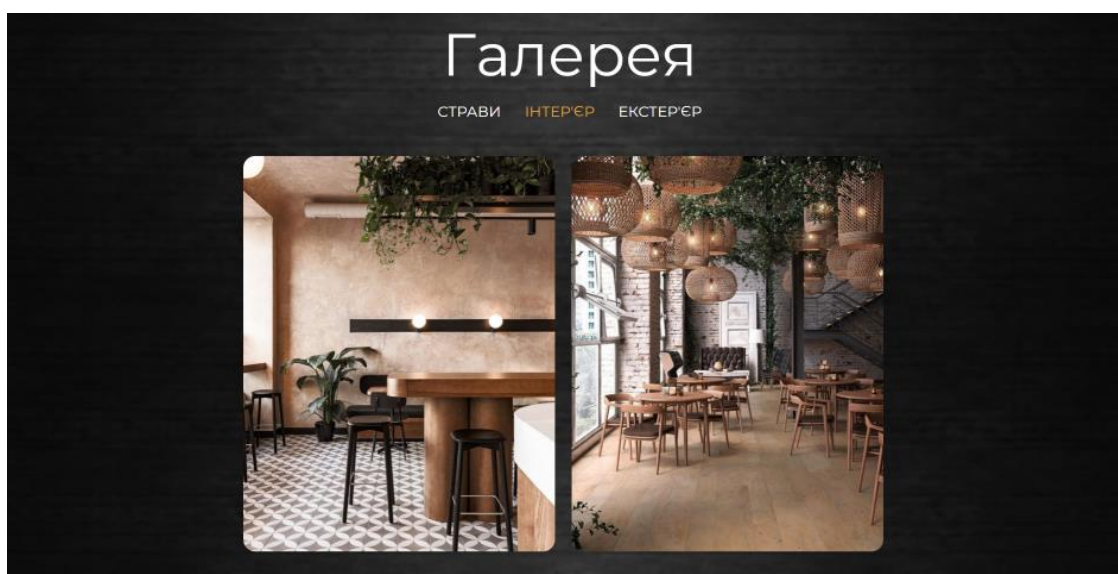


Рис. 3.9. Галерея з навігаційною панеллю

Користувач має можливість читати відгуки клієнтів, які побували у ресторані та залишати свої (рис. 3.10).

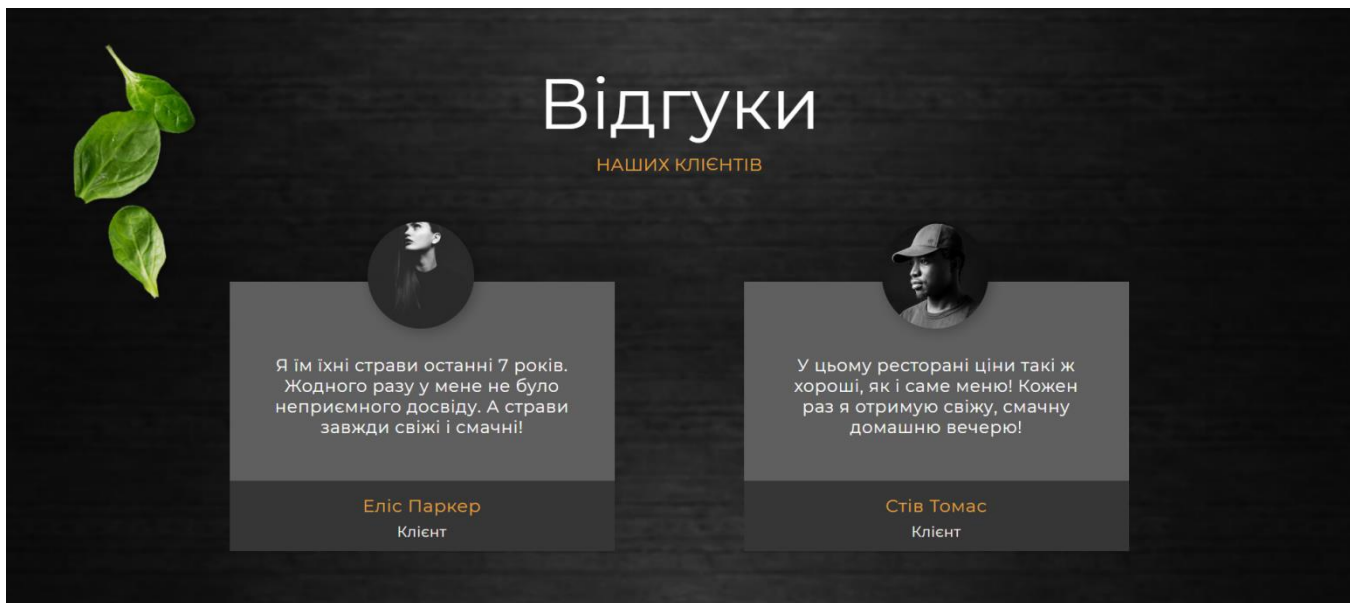


Рис. 3.10. Розділ з відгуками

Користувач має можливість підписки, щоб отримувати ексклюзивні рецепти від шеф-кухаря. Є форма, яку треба заповнити (рис. 3.11).

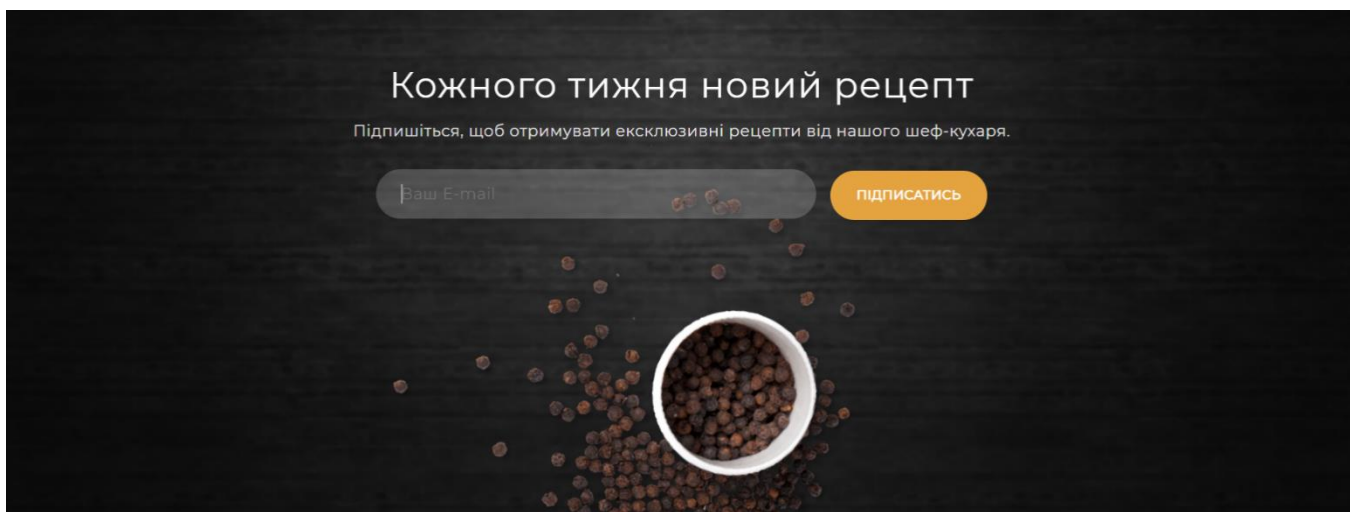


Рис. 3.11 Форма для підписки

Якщо email введений неправильно, з'явиться підказка (рис. 3.12).

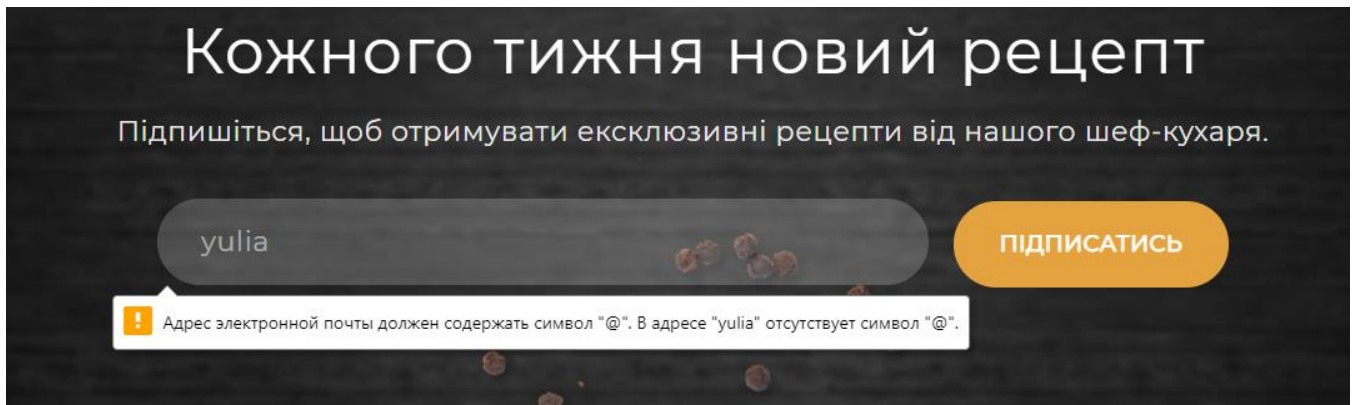


Рис. 3.12 Підказка

Після підписки, користувачу впливає повідомлення про успішну підписку (рис. 3.13).

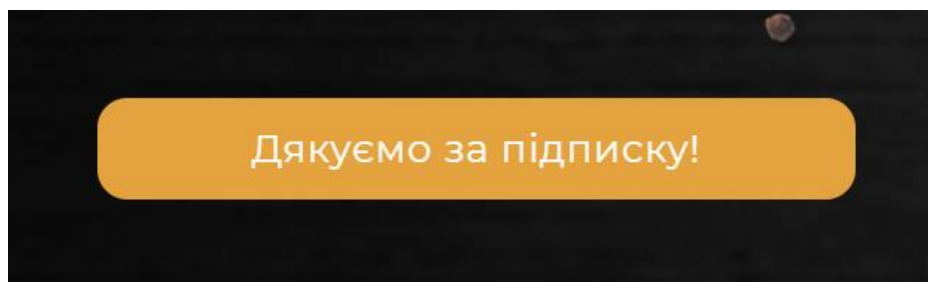


Рис. 3.13. Модальне вікно підтвердження підписки

В цей час, адміну прийде лист на пошту з даними користувача, який підписався (рис. 3.14).

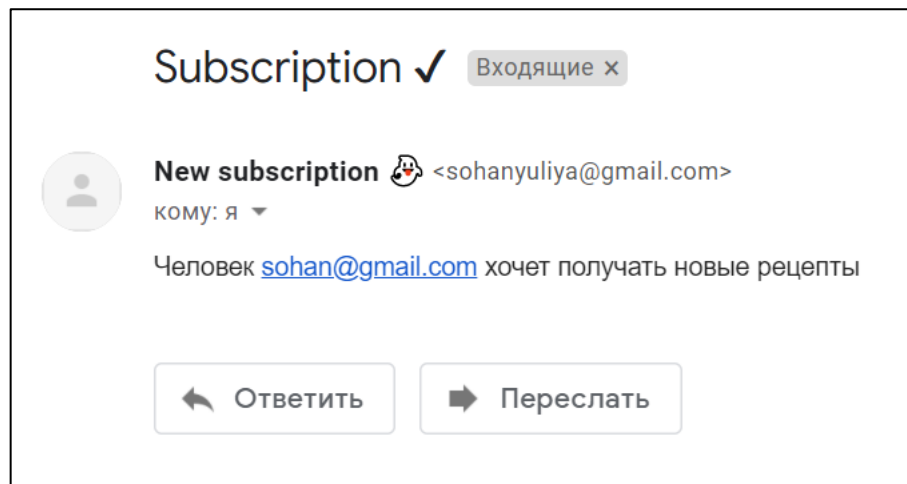


Рис. 3.14. Лист з даними користувача

Останній розділ сайту це підвал. В ньому розміщені соціальні мережі ресторану та навігаційна панель, для зручного переміщення по розділам сайту(рис. 3.15).

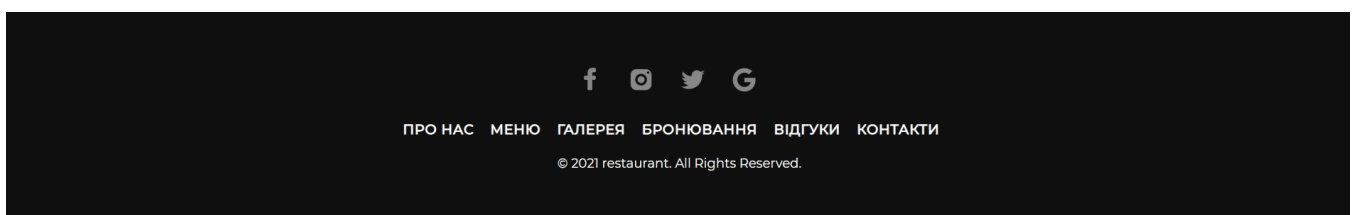


Рис. 3.15. Підвал

3.1. Загальний опис роботи веб-сайту

Технологія використання веб-сайту полягає в наступному: користувач заходить на сайт, переглядає інформацію про ресторан, відгуки, галерею ресторану, а також меню та ціни на страви. За бажанням, бронює стіл на обрану дату та час. Після бронювання очікує підтвердження. Після цього, за бажанням підписується, щоб отримувати ексклюзивні рецепти від шеф-кухаря.

Висновки до розділу 3

В результаті виконаної роботи було розроблено веб-сайт ресторану по попередньо зробленому макету , який надає можливість, переглядати інформацію про ресторан, його контактні данні, місце знаходження, години роботи, опис самого ресторану, його меню, ціни на страви, бронювати стіл, переглядати галерею, читати відгуки та підписуватись, щоб отримувати ексклюзивні рецепти від шеф-кухаря.

ВИСНОВКИ

В результаті виконаної роботи було розроблено веб-сайт ресторану, який надає можливість, переглядати інформацію про ресторан, його контактні данні, місце знаходження, години роботи, опис самого ресторану, його меню, ціни на страви, бронювати стіл, переглядати галерею, читати відгуки та підписуватись, щоб отримувати ексклюзивні рецепти від шеф-кухаря.

Проаналізовано існуючі веб-сати, визначенно їх переваги та недоліки. Проаналізовано вимоги до нового веб-сайту рестарана.

З використанням бібліотеки React.js та мови програмування JavaScript реалізовано графічний веб-сайт ресторану. Для взаємодії з клієнтами за допомогою API було створено Node.js сервер.

Завдяки тому, що веб-сайт написан за допомогою сучасної мови програмування та її провідних бібліотек, гарантується з часом підтримка всіх компонентів та їх подальша модернізація. Даний сайт значно спрощує процес пошуку інфоормачії про цей ресторан.

Реалізовані можливості програмного продукту повністю задовільняють поставленій задачі.

СПИСОК БІБЛЮГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. HTMLBOOK - Для тех, кто делает сайты [Електронний ресурс]. — Режим доступу: <http://htmlbook.ru/>(дата звернення 15.05.2021) — Назва з екрану.
2. Большая книга CSS3 [Електронний ресурс]. — Режим доступу: [ftp://ftp.micronet-rostov.ru/linux-support/books/programming/HTML-CSS/Дэвид Сойер Макфарланд — Большая книга CSS3/](ftp://ftp.micronet-rostov.ru/linux-support/books/programming/HTML-CSS/Дэвид%20Сойер%20Макфарланд%20—%20Большая%20книга%20CSS3/)(дата звернення 15.05.2021) — Назва з екрану.
3. Choosing a default language [Електронний ресурс]. — Режим доступу: <https://confluence.atlassian.com/adminjiraserver071/choosing-a-default-language-802592304/>(дата звернення 15.05.2021) — Назва з екрану.
4. Выразительный Javascript [Електронний ресурс]. — Режим доступу: https://karmazzin.gitbooks.io/eloquentjavascript_ru/content/(дата звернення 18.05.2021) — Назва з екрану.
5. Современные возможности ES-2015 [Електронний ресурс]. — Режим доступу: <https://learn.javascript.ru/es-modern/>(дата звернення 18.05.2021) — Назва з екрану.
6. The MIT License (MIT) [Електронний ресурс]. — Режим доступу: <https://opensource.org/licenses/MIT/>(дата звернення 15.05.2021) — Назва з екрану.
7. March 2013 Web Server Survey [Електронний ресурс]. — Режим доступу: <http://news.netcraft.com/archives/2013/03/01/march-2013-web-serversurvey/>(дата звернення 15.05.2021) — Назва з екрану.
8. What is REST [Електронний ресурс]. — Режим доступу: <http://www.restapitutorial.com/lessons/whatisrest/>(дата звернення 15.05.2021) — Назва з екрану.

ДОДАТКИ

Додаток А

```
<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="utf-8" />

    <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />

    <link href='https://fonts.googleapis.com/css?family=Montserrat' rel='stylesheet'>

    <meta name="viewport" content="width=device-width, initial-scale=1" />

    <meta name="theme-color" content="#000000" />

    <meta

      name="description"

      content="Web site created using create-react-app"

    />

    <title>Simple</title>

  </head>

  <body>

    <noscript>You need to enable JavaScript to run this app.</noscript>

    <div id="root"></div>

  </body>

</html>
```

```
import React from 'react';

import ReactDOM from 'react-dom';
```

```
import './index.css';  
import App from './components/App/App';  
ReactDOM.render(  
  <React.StrictMode>  
    <App />  
  </React.StrictMode>,  
  document.getElementById('root')  
);
```

```
.feedback {  
  &__inner {  
    width: 1266px;  
    margin: 0 auto;  
    display: flex;  
    align-items: center;  
    justify-content: center;  
    flex-direction: column;  
  }  
  &__anchor {  
    margin-left: auto;  
    width: 200px;  
    margin-top: 150px;  
    margin-right: 100px;  
  }  
}
```

```
&__content {  
display: flex;  
justify-content: space-between;  
width: 100%;  
}
```

```
&__title-container {  
margin-bottom: 50px;  
margin-top: 150px;  
}
```

```
&__title {  
font-size: 100px;  
font-style: normal;  
font-weight: 400;  
line-height: 122px;  
letter-spacing: 0;  
text-align: left;  
}
```

```
&__subtitle {  
font-size: 25px;  
font-style: normal;  
font-weight: 500;  
line-height: 30px;  
letter-spacing: 0;  
text-align: center;
```

```
color: rgba(229, 163, 63, 1);
```

```
.booking {
```

```
  &__anchor {
```

```
    margin-left: auto;
```

```
    width: 200px;
```

```
    margin-top: 100px;
```

```
    margin-right: 100px;
```

```
  }
```

```
&__title-container {
```

```
  width: 930px;
```

```
  display: flex;
```

```
  justify-content: center;
```

```
  align-items: center;
```

```
  flex-direction: column;
```

```
  margin: 100px auto 50px;
```

```
}
```

```
&__title {
```

```
  font-size: 85px;
```

```
  font-style: normal;
```

```
  font-weight: 500;
```

```
  line-height: 122px;
```

```
  letter-spacing: 0.03em;
```

```
text-align: center;
```

```
}
```

```
&__subtitle {
```

```
font-size: 25px;
```

```
font-style: normal;
```

```
font-weight: 500;
```

```
line-height: 30px;
```

```
letter-spacing: 0.03em;
```

```
text-align: center;
```

```
color: #E5A33F;
```

```
}
```

```
}
```

```
.form {
```

```
width: 1240px;
```

```
margin: 0 auto;
```

```
&__control, input, .css-yk16xz-control, .css-1pahdxg-control, .css-26l3qy-menu,  
&__textarea {
```

```
width: 363px;
```

```
height: 68px;
```

```
background: rgba(196, 196, 196, 0.21);
```

```
border-radius: 20px;
```

```
font-size: 22px;

font-style: normal;

font-weight: 400;

line-height: 27px;

letter-spacing: 0.03em;

color: rgba(255, 255, 255, 0.7);

border: none;

&:focus {

    outline: none;

}

margin-right: 50px;

margin-bottom: 20px;

&:nth-child(3) {

    margin-right: 0px;

}

}

&__control, input, &__textarea {

    padding-left: 10px;

}

&__textarea {

    margin-right: 0px !important;

}

.css-g1d714-ValueContainer {

    width: 363px;
```

```
height: 68px;

font-size: 22px;

font-style: normal;

font-weight: 400;

line-height: 30px;

letter-spacing: 0.03em;

text-align: center;

color: rgba(255, 255, 255, 0.7);

&:focus {

    outline: none;

}

}

&__group {

    display: flex;

    flex-wrap: wrap;

}

&__textarea {

    width: 1194px !important;

    height: 120px !important;

    padding-top: 5px;

}

&__submit {

    width: 1194px;

    height: 75px;
```



```
background: #E5A33F;  
border-radius: 20px;  
border: none;  
font-size: 18px;  
font-style: normal;  
font-weight: 700;  
line-height: 22px;  
letter-spacing: 0.03em;  
text-align: center;  
text-transform: uppercase;  
}
```

```
.css-4ljt47-MenuList {  
background-color: #464646;  
border-radius: 25px;  
}
```

```
.css-1uccc91-singleValue {  
color: rgba(255, 255, 255, 0.7);  
position: static;  
margin-top: 10px;  
}
```

```
.css-9gakcf-option {  
background-color: #262626;  
}
```

```
}  
  
.footer {  
  &__top {  
    width: 950px;  
    margin: 150px auto 0;  
  }  
  
  &__title-container {  
    margin-bottom: 40px;  
  }  
  
  &__title {  
    font-size: 50px;  
    font-style: normal;  
    font-weight: 500;  
    line-height: 61px;  
    letter-spacing: 0.03em;  
    text-align: center;  
  }  
  
  &__subtitle {  
    font-size: 22px;  
    font-style: normal;  
    font-weight: 300;  
    line-height: 27px;  
    letter-spacing: 0.03em;  
    text-align: center;
```

```
color: rgba(255, 255, 255, 0.9);
    margin-top: 20px;
}
&__subscription {
    display: flex;
    justify-content: center;
    margin-bottom: 425px;
}
&__form-control {
    width: 617px;
    height: 69px;
    background: rgba(255, 255, 255, 0.2);
    border-radius: 500px;
    border: none;
    font-size: 22px;
    font-style: normal;
    font-weight: 400;
    line-height: 27px;
    letter-spacing: 0.03em;
    text-align: left;
    padding-left: 35px;
    color: rgba(255, 255, 255, 0.52);
    margin-right: 20px;
    &:focus {
```

outline: none;

}

}

&__submit {

width: 220px;

height: 69px;

background: #E5A33F;

border-radius: 500px;

border: none;

font-size: 18px;

font-style: normal;

font-weight: 700;

line-height: 22px;

letter-spacing: 0.03em;

}

&__bottom {

width: 100%;

background-color: rgba(15, 15, 15, 1);

display: flex;

flex-direction: column;

align-items: center;

li, a {

text-decoration: none;

```
list-style-type: none;
}
}
&__socials {
display: flex;
margin-top: 82px;
margin-bottom: 36px;
img {
margin-right: 32px;
}
}
&__nav {
margin-bottom: 25px;
ul {
display: flex;
li {
margin-right: 23px;
font-size: 19px;
font-style: normal;
font-weight: 700;
line-height: 23px;
letter-spacing: 0.05em;
text-align: left;
}
}
```

```
}  
  
}  
  
&__rights {  
    margin-bottom: 80px;  
    font-size: 18px;  
    font-style: normal;  
    font-weight: 400;  
    line-height: 22px;  
    letter-spacing: 0;  
    text-align: left;  
}  
}
```

```
.gallery {  
    .menu__top {  
        margin-top: 100px;  
    }  
  
    &__anchor {  
        margin-left: auto;  
        width: 200px;  
        margin-top: 150px;  
        margin-right: 100px;  
    }  
  
    &__content {
```

```
width: 1100px;  
display: flex;  
margin: 0 auto;  
img {  
    height: 665px;  
    width: 525px;  
    margin-right: 28px;  
    &:last-child {  
        margin-right: 0;  
    }  
}  
}  
}  
  
.header {  
    display: flex;  
    justify-content: space-between;  
    font-size: 25px;  
    line-height: 30px;  
    &__left {  
        a {  
            margin: 10px 40px !important;  
        }  
    }  
}
```

```
&__left, &__right {  
    display: flex;  
    padding: 65px 30px;  
    a , svg{  
        margin: 10px;  
        cursor: pointer;  
    }  
    svg {  
        margin-left: 20px;  
    }  
    a {  
        font-weight: 500;  
        text-decoration: none;  
    }  
}  
  
.contacts {  
    margin-right: 30px;  
    width: 300px;  
    height: 168px;  
    background: rgba(196, 196, 196, 0.21);  
    box-shadow: 0px 4px 4px rgba(0, 0, 0, 0.25);  
    border-radius: 20px;  
    display: flex;
```



```
flex-direction: column;  
align-items: center;  
svg {  
    width: 50px !important;  
    height: 50px !important;  
    margin-top: 30px;  
    margin-bottom: 20px;  
    path {  
        fill: #E5A33F  
    }φ  
}  
&__title {  
    font-size: 18px;  
    font-style: normal;  
    font-weight: 600;  
    line-height: 22px;  
    letter-spacing: 0.03em;  
    text-align: center;  
    margin-bottom: 7px;  
}  
&__subtitle {  
    font-size: 15px;  
    font-style: normal;  
    font-weight: 400;
```

```
line-height: 18px;
letter-spacing: 0.03em;
text-align: center;
}
}
import rating from '../assets/rating.png';
import desert1 from '../assets/desert1.png';
import desert2 from '../assets/desert2.png';
import desert3 from '../assets/desert3.png';
import desert4 from '../assets/desert4.png';
import main1 from '../assets/main1.png';
import main2 from '../assets/main2.png';
import main3 from '../assets/main3.png';
import main4 from '../assets/main4.png';
import main5 from '../assets/main5.png';
import main6 from '../assets/main6.png';
import main7 from '../assets/main7.png';
import main8 from '../assets/main8.png';
import cock1 from '../assets/cock1.png';
import cock2 from '../assets/cock2.png';
import cock3 from '../assets/cock3.png';
import cock4 from '../assets/cock4.png';
import snack1 from '../assets/snacks1.png';
import snack2 from '../assets/snack2.png';
```

```
import snack3 from '../assets/snack3.png';  
import snack4 from '../assets/snack4.png';  
import milk1 from '../assets/milk1.png';  
import milk2 from '../assets/milk2.png';  
import milk3 from '../assets/milk3.png';  
import milk4 from '../assets/milk4.png';
```

```
export const menuData = [  
  {  
    category: 'main-dishes',  
    name: 'Креветки у соусі',  
    image: main1,  
    price: '180',  
    rating: rating,  
    description: 'З соусом Цітронет, базиліком, аспарагусом і сиром Пармезан.',  
  },  
  {  
    category: 'main-dishes',  
    image: main2,  
    name: 'Устриці',  
    price: '150',  
    rating: rating,  
    description: 'З манго, авокадо і цитрусовим соусом.',  
  },
```

```
{  
  category: 'main-dishes',  
  image: main3,  
  name: 'Морепродукти',  
  price: '130',  
  rating: rating,  
  description: 'Подається з вершковим соусом і хлібом Чіабатта.',  
},  
  
{  
  category: 'main-dishes',  
  image: main4,  
  name: 'Фуа-гра',  
  price: '130',  
  rating: rating,  
  description: 'З печеними яблуками і ягідним соусом .',  
},  
  
{  
  category: 'main-dishes',  
  image: main5,  
  name: 'Крем-суп',  
  price: '130',  
  rating: rating,  
  description: 'З болгарським перцем та беконом і цвітною капустою .',  
},
```

```
{
  category: 'main-dishes',
  image: main6,
  name: 'Гаспачо',
  price: '130',
  rating: rating,
  description: 'З качкою в соєвому бульйоні з пекінською капустою і яйцем.',
},
{
  category: 'main-dishes',
  image: main7,
  name: 'Салат з лосося',
  price: '130',
  rating: rating,
  description: 'З сирним сиром, часником і хріном, овочами і пелюстками мигдалю на Бородінському хлібі.',
},
{
  category: 'main-dishes',
  image: main8,
  name: 'Реберця',
  price: '130',
  rating: rating,
  description: 'З овочами гриль і соусом з чорного перцю.'
```

```
},  
  
{  
  category: 'snacks',  
  image: snack1,  
  name: 'Піта з овочами',  
  price: '130',  
  rating: rating,  
  description: 'З запеченим перцем з соусом Горгонзола.' },  
  
{  
  category: 'snacks',  
  image: snack2,  
  name: 'Начос з соусами',  
  price: '130',  
  rating: rating,  
  description: 'З сиром Страчателла та гуакамоле.',  
},  
  
{  
  category: 'snacks',  
  image: snack3,  
  name: 'Брускети',  
  price: '130',  
  rating: rating,  
  description: 'З кроликом і в'яленими томатами.',  
},
```

```
{  
  category: 'snacks',  
  image: snack4,  
  name: 'Acopmi',  
  price: '130',  
  rating: rating,  
  description: 'З шпинатом, сиром Рікота та червоною ікрою у вершковому соусі.',  
},  
{  
  category: 'cocktails',  
  image: cock1,  
  name: 'Blueberry Thym G&T',  
  price: '130',  
  rating: rating,  
  description: 'У складі коктейлю лімончелло, сорбет з естрагону і горілка, манго і сік вишні',  
},  
{  
  category: 'cocktails',  
  image: cock2,  
  name: 'Little Love',  
  price: '130',  
  rating: rating,
```

description: 'У складі коктейлю віскі Jack Daniels Old № 7, маракуйя, лимонний фреш.'

},

{

category: 'cocktails',

image: cock3,

name: 'White rabbit',

price: '130',

rating: rating,

description: 'У складі коктейлю американський бурбон, карибська маракуйя та кисло-солодкий мікс.'

},

{

category: 'cocktails',

image: cock4,

name: 'Rye Manhattan',

price: '130',

rating: rating,

description: 'У складі коктейлю житній віскі, червоний вермут, херес, біттери.'

},

{

category: 'cocktails',

image: milk1,

name: 'Коктейль з полуницею',


```
price: '130',  
  rating: rating,  
  description: 'Молоко, морозиво вершкове, цукровий сироп, полуниця.',  
},  
{  
  category: 'cocktails',  
  image: milk2,  
  name: 'Молочный коктейль',  
  price: '130',  
  rating: rating,  
  description: 'Молоко, морозиво вершкове, цукровий сироп.',  
},  
{  
  category: 'cocktails',  
  image: milk3,  
  name: 'Коктейль з вафлями',  
  price: '130',  
  rating: rating,  
  description: 'Молоко, морозиво вершкове, цукровий сироп, вафлі.',  
},  
{  
  category: 'cocktails',  
  image: milk4,  
  name: 'Коктейль з шоколадом',
```

```
price: '130',  
  rating: rating,  
  description: 'Молоко, морозиво вершкове, цукровий сироп, шоколад.',  
},  
{  
  category: 'desert',  
  image: desert1,  
  name: 'Макаруни',  
  price: '130',  
  rating: rating,  
  description: 'З крем-сиром Маскарпоне, лікером Бейліс, малиною та полуницею.',  
},  
{  
  category: 'desert',  
  image: desert2,  
  name: 'Морозиво',  
  price: '130',  
  rating: rating,  
  description: 'З фруктами та шоколадними крихтами.',  
},  
{  
  category: 'desert',  
  image: desert3,  
  name: 'Фондан',
```

```
price: '130',  
  rating: rating,  
  description: 'З ванільним морозивом в фісташках і ягідним джемом.',  
},  
{  
  category: 'desert',  
  image: desert4,  
  name: 'Чізкейк',  
  price: '130',  
  rating: rating,  
  description: 'З сиром Рікота та маракуєю.',  
},  
.menu {  
  margin-top: 350px;  
  &__anchor {  
    margin-left: auto;  
    width: 200px;  
    margin-right: 100px;  
  }  
  
  &__inner {  
    width: 1337px;  
    margin: 100px auto 0;  
  }  
}
```

```
&__title {  
    font-size: 100px;  
    font-style: normal;  
    font-weight: 400;  
    line-height: 122px;  
    letter-spacing: 0em;  
    text-align: left;  
}
```

```
&__top {  
    display: flex;  
    width: 100%;  
    justify-content: center;  
    align-items: center;  
    flex-direction: column;  
}
```

```
&__nav-group {  
    display: flex;  
    margin-bottom: 40px;  
}
```

```
&__nav-item {  
    font-size: 25px;  
    font-style: normal;  
    font-weight: 500;
```

line-height: 30px;

letter-spacing: 0em;

text-align: left;

cursor: pointer;

list-style-type: none;

margin: 20px;

&__selected {

color: #E5A33F;

}

}

&__content {

display: flex;

flex-wrap: wrap;

}

&__item-content {

padding: 12px;

}

&__item-price-container {

display: flex;

justify-content: space-between;

margin: 10px 0;

```
}  
  
  &__item {  
    background-color: #414141;  
    border-radius: 20px;  
    width: 293px;  
    height: 393px;  
    margin-right: 55px;  
    margin-bottom: 50px;  
    &:nth-child(4) {  
      margin-right: 0;  
    }  
    &:last-child {  
      margin-right: 0;  
    }  
  }  
  
  &__item-price {  
    font-size: 18px;  
    font-style: normal;  
    font-weight: 600;  
    line-height: 22px;  
    letter-spacing: 0.03em;  
    text-align: left;  
  }
```

```
&__item-main-image {  
    width: 293px;  
    height: 240px;  
}
```

```
&__item-title {  
    font-size: 20px;  
    font-style: normal;  
    font-weight: 600;  
    line-height: 24px;  
    letter-spacing: 0.03em;  
    text-align: left;  
}
```

```
&__item-description {  
    font-size: 14px;  
    font-style: normal;  
    font-weight: 300;  
    line-height: 17px;  
    letter-spacing: 0.03em;  
    text-align: left;  
}
```

```
}  
  
.snackbar {
```

```
position: fixed;  
bottom: 10%;  
left: 5%;  
z-index: 9999;  
background: #E5A33F;  
width: 509px;  
height: 68px;  
border-radius: 20px;  
display: flex;  
justify-content: center;  
align-items: center;  
p {  
    font-size: 26px;  
    font-style: normal;  
    font-weight: 500;  
    line-height: 32px;  
    letter-spacing: 0.03em;  
    text-align: center;  
}  
}  
* {  
    font-family: 'Montserrat', serif;  
    color: #fff;  
    padding: 0;
```



```
margin: 0;

box-sizing: border-box;
}

{
  "name": "simple",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@fortawesome/fontawesome-svg-core": "^1.2.35",
    "@fortawesome/free-solid-svg-icons": "^5.15.3",
    "@fortawesome/react-fontawesome": "^0.1.14",
    "@testing-library/jest-dom": "^5.11.4",
    "@testing-library/react": "^11.1.0",
    "@testing-library/user-event": "^12.1.10",
    "body-parser": "^1.19.0",
    "classnames": "^2.3.1",
    "cors": "^2.8.5",
    "dotenv": "^10.0.0",
    "express": "^4.17.1",
    "express-favicon": "^2.0.1",
    "formik": "^2.2.8",
    "node-sass": "4.14.1",
    "nodemailer": "^6.6.0",
    "path": "^0.12.7",
```

```

"react": "^17.0.2",
  "react-datepicker": "^3.8.0",
  "react-dom": "^17.0.2",
  "react-input-mask": "^2.0.4",
  "react-scripts": "4.0.3",
  "react-select": "^4.3.1",
  "web-vitals": "^1.0.1"
},
"scripts": {
  "start": "react-scripts start",
  "build": "react-scripts build",
  "test": "react-scripts test",
  "eject": "react-scripts eject"
},
"proxy": "http://localhost:5000/",
"eslintConfig": {
  "extends": [
    "react-app",
    "react-app/jest"
  ]
},
"browserslist": {
  "production": [
    ">0.2%",

```

```
"not dead",  
  "not op_mini all"  
],  
"development": [  
  "last 1 chrome version",  
  "last 1 firefox version",  
  "last 1 safari version"  
]  
}  
}
```